Recurrent neural networks for spatiotemporal prediction of chaotic dynamics

Vlachas Pantelis

Computational Science and Engineering Lab
ETH Zürich
With:

Petros Koumoutsakos - ETH Zurich, CSE Lab
Wonmin Byeon - NVIDIA, Zhong Win Wan - MIT, Themistoklis Sapsis - MIT
Jaideep Pathak, Brian Hunt, Edward Ott - Univ. of Maryland
Motivation

High-dimensional systems with chaotic behaviour are abound in nature and engineering.

- Ocean currents
- N body problems in astrophysics
- Climate

Forecasting of such systems is of great practical importance in nature and engineering.
Recurrent Neural Networks

Model $o_t+1 = o_t + \Delta t \dot{o}_t$

with $\dot{o}_t = f(o_t, o_{t-1}, o_{t-2}, \ldots)$

history encoded in $h_{t-1}$
Training Data - Reducing the dimensionality (observable)

High dimensional

SVD
Singular Value Decomposition (SVD)

Training Data!

Low dimensional state (most energetic modes)

Throw away modes with low energy

Cumulative Energy in %

\[
\int = 100\% \\
97\% of total Energy
\]

Modes

0
25
50
75
100

\[ \nu = 1/10 \]

20 Modes (observable)
How to train these networks?

Time-series: \(\{\tilde{o}_1, \tilde{o}_2, \tilde{o}_3, \ldots, \tilde{o}_{11}, \tilde{o}_{12}\}\)

Target: \(\{\tilde{o}_1, \tilde{o}_2, \tilde{o}_3, \ldots, \tilde{o}_{11}\}\) or \(\{\tilde{o}_2, \tilde{o}_3, \tilde{o}_4, \ldots, \tilde{o}_{12}\}\)

Algorithm: Back-propagation through time BPPT

\[L = \frac{1}{2} | |o_4 - \tilde{o}_4||_2 + | |o_5 - \tilde{o}_5||_2\]
Forecasting on UNSEEN data - Iterative prediction in practice $d_h=100$, $d=10$

1. Train the model with BPTT*

2. How to predict on Test (unseen) data set?

Short term history known

*Back-propagation through time BPPT
Forecasting - Iterative prediction in practice $d_h=100$, $d=10$

Prediction in the **reduced** space

Expanded prediction in the **original** space

$v = 1/10$

$v = 1/16$
Problems?

Iterative prediction causes **accumulation of prediction errors**. Especially in reduced order observables, it may cause **divergence**.

- Dynamics underrepresented in training data
- Scarce data in attractor boundaries

- Under-resolved high dimensional dynamics
- Models not generalising
Capturing Long-Term Behavior

**PROBLEMS**

- Iterative forecasting with LSTM suffers from **accumulation** of prediction **errors**
- There is no mechanism that guarantees that trajectories remain on the attractor

**SOLUTION - MEAN STOCHASTIC MODEL (MSM)**

- Ornstein-Uhlenbeck process - computationally cheap
- Converges in the **long-term in mean statistical behavior**
- Relies on **global attractor statistics**
- Efficient in highly chaotic systems

Hybrid LSTM - MSM approach: Use **MSM** in regions underrepresented in the training data or near attractor **boundaries**

\[
\begin{align*}
dz_t &= c z_t \, dt + \zeta \, dW_t \\
\zeta &= \sqrt{-2 \, c \, \sigma_z} \\
c &= \frac{1}{T}
\end{align*}
\]
### Results - Kuramoto-Sivashinsky - Comparison with Gaussian Process Regression (GPR)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V$</td>
<td>Total number of initial conditions ($i_c$)</td>
</tr>
<tr>
<td>$k$</td>
<td>Mode number</td>
</tr>
<tr>
<td>$i$</td>
<td>IC index</td>
</tr>
<tr>
<td>$z^i_k$</td>
<td><strong>True</strong> state of mode $k$ starting from $i_c$ $i$</td>
</tr>
<tr>
<td>$\tilde{z}^i_k$</td>
<td><strong>Predicted</strong> state of mode $k$ starting from $i_c$ $i$</td>
</tr>
</tbody>
</table>

Root mean square error (RMSE)

$$RMSE(z_k) = \sqrt{1/V \sum_{i=1}^{V} (z^i_k - \tilde{z}^i_k)^2}$$

Time evolution of the RMSE of the **most energetic** PCA mode over 1000 initial conditions
Problems?

1. Iterative prediction causes **accumulation of prediction errors**. Especially in reduced order observables, it may cause **divergence**.
   - Dynamics underrepresented in training data
   - Scarce data in attractor boundaries
   - Under-resolved high dimensional dynamics
   - Insufficient training, low quality data, noisy

2. **Vanishing gradients problem during training**: As the gradient is back-propagated during training of the networks it may **vanish to zero or explode**.

**Solution?** Hybrid LSTM - MSM approach

**Solution?** Sophisticated architectures
RNN Models

- **LSTM Cell**
  - $g^i_t$ and $g^o_t$ are gates.
  - $\sigma$ denotes the sigmoid function.
  - $\tanh$ represents the hyperbolic tangent function.
  - $W$ denotes the weight matrix.

- **GRU Cell**
  - $r_t$ and $z_t$ are update gates.
  - $\sigma$ denotes the sigmoid function.
  - $\tanh$ represents the hyperbolic tangent function.

- **Reservoir Computer**
  - $W \sim$ denotes random weight initialization.
  - $\tanh (W \cdot)$ represents a weighted sum followed by hyperbolic tangent.

- **Unitary**
  - $\modRelu$ represents modReLU activation function.
  - $h_{t-1} \in \mathbb{R}^d_h$ denotes the hidden state.
Lorenz-96 - 35/40 mode observable

Diagram showing time evolution of different models (GRU, LSTM, RC-6000, RC-18000, Unit) with NRMSE and color-coded error metrics.
Benchmarking - Lorenz-96 - 35/40 SVD mode observable

- Reduced-order $d_o = 35$
- Full-state $d_o = 40$

![Graph showing VPT for different models (RC, GRU, LSTM, Unit) for reduced-order and full-state configurations across varying RAM memory.](image-url)

- **FULL-STATE $d_o = 40$**
- RAM Memory [MB] vs. VPT

- RC, GRU, LSTM, Unit
Lorenz-96 - full state information + parallelism
Kuramoto-Sivashinsky - Full state information + Parallelism

TARGET

RC-1000
RC-3000
GRU-80
LSTM-80

NRMSE

TARGET

RC-1000 NRMSE
RC-3000 NRMSE
GRU-80 NRMSE
LSTM-80 NRMSE

o

ot

{RNN}

G

z

G

z

I

{z}

I

{z}

do

Kuramoto-Sivashinsky - Full state information + Parallelism
More applications...

1. Lorenz 96

2. Barotropic Model

3. Kuramoto Sivashinsky

4. Deep Reinforcement Learning

\[ Q^*(s, \alpha) = E_{\gamma} \left[ \sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_t = s_1, s_{t-1} = s_2, \ldots, \alpha_t = \alpha_1, \alpha_{t-1} = \alpha_2, \ldots \right] \approx Q^*(s, \alpha) \]

Lyapunov Spectrum
Summary & Future Outlook

- **LSTMs** utilized as **nonlinear** data-driven predictors of high dimensional chaotic dynamical systems
- Coupled with **Mean Stochastic Model (MSM)** to capture **long-term statistics**
- Their prediction **accuracy** was benchmarked against:
  - Gaussian Processes (GPR), Reservoir Computers (RC), GRUs, Unitary RNNs **ongoing work**...
  - in Kuramoto-Sivashinsky, Lorenz-96 system, Barotropic climate model
- Dynamical system surrogates - **Lyapunov spectrum** - employed in **Reinforcement Learning**
- Other open questions
  - Stochastic (bayesian) RNNs estimating uncertainty of forecasts ?
  - Training procedures ?